

SPRINT 8.2 PYTHON EN PBI

Esta tarea consiste en la elaboración de un informe de Power BI, aprovechando las capacidades analíticas de Python. Se utilizarán los scripts de Python creados previamente en la Tarea 1 para generar visualizaciones personalizadas con las bibliotecas Seaborn y Matplotlib. Estas visualizaciones serán integradas en el informe de Power BI para ofrecer una comprensión más profunda de la capacidad del lenguaje de programación en la herramienta Power BI.

Pasos para conectar Python y PBI

En primer lugar, abrimos PBI y seleccionamos “Informe en blanco”. Después, clicamos en “Obtener datos desde otro origen” y en la barra de búsqueda elegimos la opción “Script en Python” y clicamos en el botón **conectar**.

Obtener datos

python

Todo

Script de Python

Otras

Conectores certificados

Aplicaciones de plantilla

Conectar

Cancelar

Power BI con Python

Cristina Cumplido Huertas

Al momento, aparecerá una pantalla donde debemos poner el siguiente código para realizar la conexión:

```
import mysql.connector as sql

mydb= sql.connect(
    user="root",
    password="cristina1234",
    database="sprint4")

import pandas as pd

companies = pd.read_sql("SELECT * FROM companies", mydb)
credit_cards = pd.read_sql("SELECT * FROM credit_cards", mydb)
card_status = pd.read_sql("SELECT * FROM card_status", mydb)
products = pd.read_sql("SELECT * FROM products", mydb)
transaction_products = pd.read_sql("SELECT * FROM
transaction_products", mydb)
transactions = pd.read_sql("SELECT * FROM transactions", mydb)
users = pd.read_sql("SELECT * FROM users", mydb)
```

Script de Python

Script

```
user="root",
password="cristina1234",
database="sprint4")
import pandas as pd

companies = pd.read_sql("SELECT * FROM companies", mydb)
credit_cards = pd.read_sql("SELECT * FROM credit_cards", mydb)
card_status = pd.read_sql("SELECT * FROM card_status", mydb)
products = pd.read_sql("SELECT * FROM products", mydb)
transaction_products = pd.read_sql("SELECT * FROM transaction_products", mydb)
transactions = pd.read_sql("SELECT * FROM transactions", mydb)
users = pd.read_sql("SELECT * FROM users", mydb)
```

El script se ejecutará con la instalación de Python siguiente: C:\Users\crist\AppData\Local\Programs\Python\Python313.
Para establecer la configuración y cambiar la instalación de Python que quiere ejecutar, vaya a Opciones y configuración.

Aceptar

Cancelar

Después, se deben cargar las tablas y hacer la relación entre ellas:

Power BI con Python

Cristina Cumplido Huertas

Navegador

Opciones de presentación ▾

Python [7]

☒

card_status

☒

companies

☒

credit_cards

☒

products

☒

transaction_products

☒

transactions

☒

users

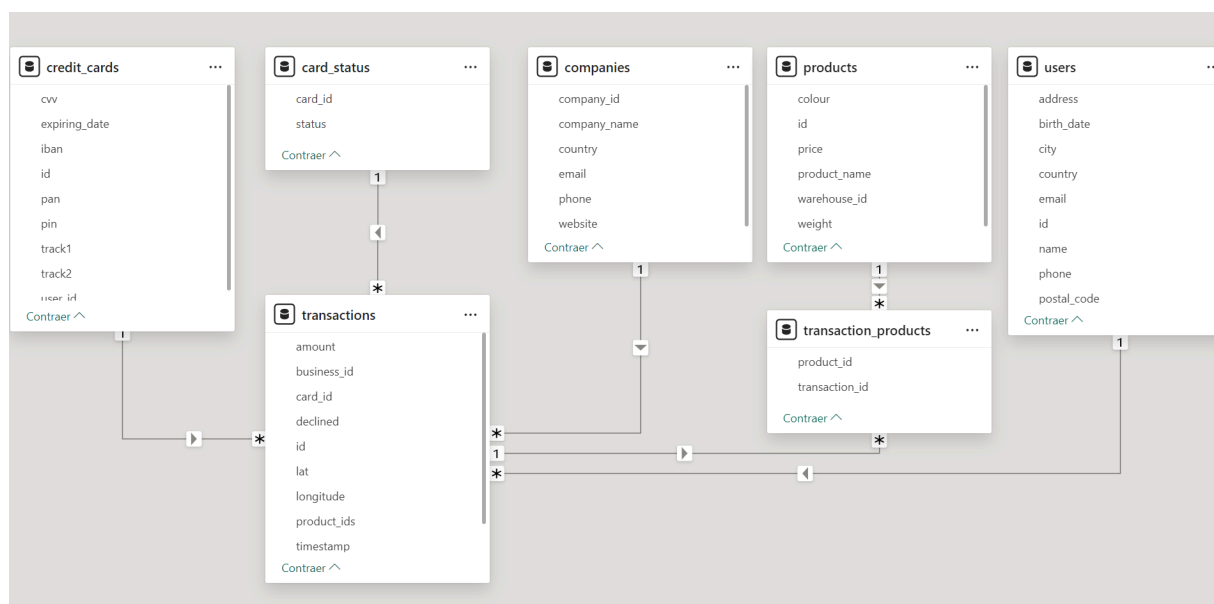
users

id	name	surname	phone	email
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonr
10	Robert	Mccarthy	(324) 746-6771	fermentum@protonmail
100	Melodie	Mclean	1-677-221-7152	risus.varius@google.ca
101	Sarah	Beck	(358) 691-4345	vitae.risus@aol.couk
102	Jasper	Landry	1-397-765-1118	consectetuer.euismod@
103	Upton	Chavez	(227) 785-6484	euismod.est@aol.ca
104	Martha	Barlow	(732) 326-5448	vulputate@hotmail.net
105	Hashim	Rose	(858) 313-6727	urna@icloud.com
106	Tanner	Valenzuela	1-346-421-3135	nascetur.ridiculus@goog
107	Victor	Valencia	(239) 569-1938	non.enim@hotmail.couk
108	Germaine	Suarez	1-931-750-6983	risus@icloud.com
109	Raven	Reynolds	(667) 453-9731	sed@aol.com
11	Joan	Baird	(981) 429-8106	et@outlook.net
110	Neil	Powers	(864) 881-6737	nec.metus@aol.edu
111	Astra	Baldwin	1-643-565-3266	adipiscing.ligula.aenean@
112	Ryder	Cole	1-572-759-8544	nec.enim.nunc@protonr
113	Risa	Frost	1-712-488-5451	neque.pellentesque@ou
114	Jasmine	Castro	1-512-143-0648	lorem@google.ca
115	Urielle	Holman	1-424-793-4354	leo@google.ca
116	Sacha	Compton	(265) 342-4775	sed.dictum.proin@yahoc
117	Halla	Pearson	(681) 698-7518	lacus.etiam@protonmail
118	Brooke	Jensen	(124) 739-9067	purus.mauris.a@icloud.c
119	Damian	Mcgee	(712) 572-8735	neque.nullam@hotmail.

Cargar

Transformar datos

Cancelar



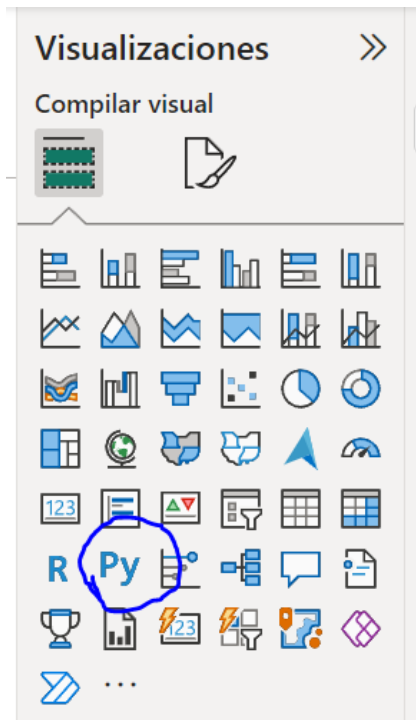
Ahora que ya tenemos las relaciones formadas podemos comenzar a graficar.

Power BI con Python

Cristina Cumplido Huertas

PBI

En primer lugar, debemos escoger en las visualizaciones el objeto visual de Python:



Después elegir los datos que vamos a usar de las tablas elegidas y después escribir y modificar nuestro código en el Editor de Scripts de Python:

Visualizaciones

Compilar visual

Filtros

Visualizaciones

Datos

Editor de scripts de Python

```
1 # Las filas duplicadas se quitarán de los datos.
2
3 # El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta y actúa como un preámbulo del script:
4 # dataset = pandas.DataFrame(amount)
5 # dataset = dataset.drop_duplicates()
6
7 # Pegue o escriba aquí el código de script:
8 import pandas as pd
9 import matplotlib.pyplot as plt
10 import seaborn as sns
11 sns.histplot(data=dataset, x='amount', kde=True)
12 plt.title('Transacciones')
13 plt.show()
```

Nivel 1

- Ejercicio 1

Variable numérica.

Inicialmente al entrar en PBI con los datos ya cargados de Python y las relaciones hechas, marcamos en el apartado DATOS la tabla “transaction” con la columna “amount” para poder graficar la frecuencia de las transacciones.

Código en Python:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
aqua_palette = ["#00FFFF", "#00CED1", "#20B2AA", "#40E0D0", "#48D1CC",
"#5F9EA0"]
sns.histplot(data=transactions, x='amount', kde=True,
color=aqua_palette[4])
plt.title("Frecuencia de las transacciones")
plt.show();
```

Código en PBI:

En el mismo editor de código de Python en PBI ya nos avisa de lo siguiente:

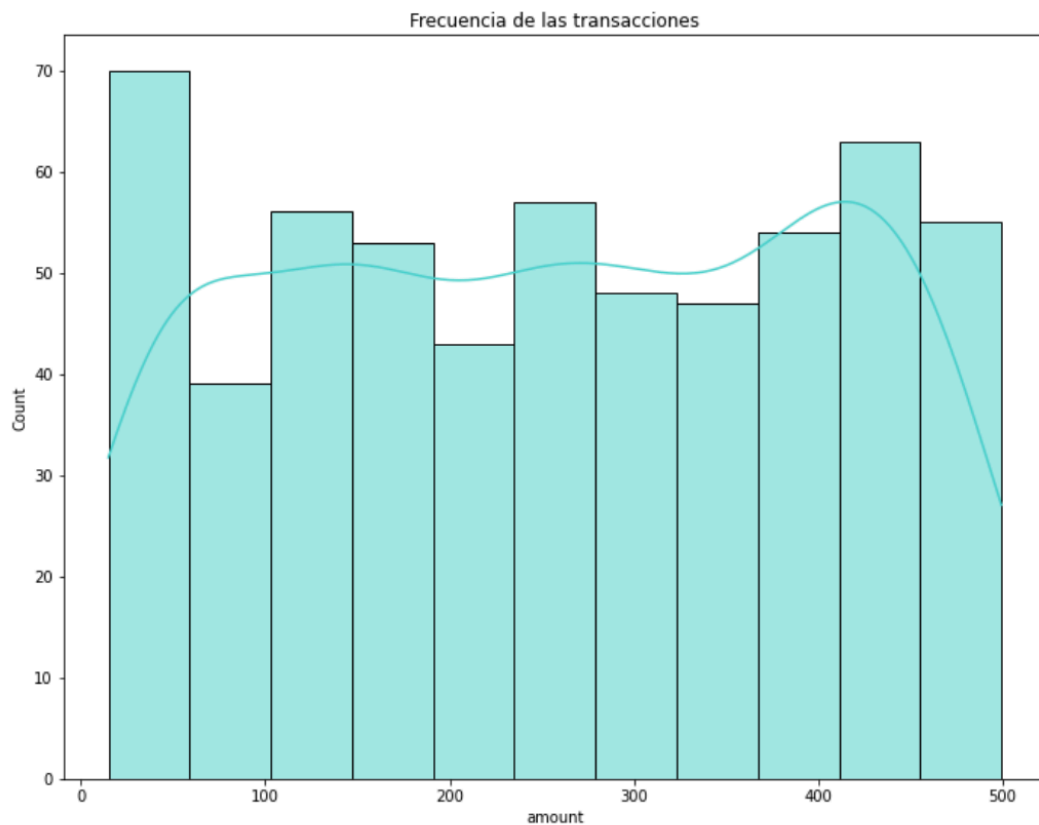
El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta y actúa como un preámbulo del script:

dataset = pandas.DataFrame(amount)

dataset = dataset.drop_duplicates()

Código modificado:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
aqua_palette = ["#00FFFF", "#00CED1", "#20B2AA", "#40E0D0", "#48D1CC",
"#5F9EA0"]
sns.histplot(data=dataset, x='amount', kde=True, color=aqua_palette[4])
plt.title("Frecuencia de las transacciones")
plt.show();
```



- Ejercicio 2

Dos variables numéricas.

Marcamos en datos "price" y "weight".

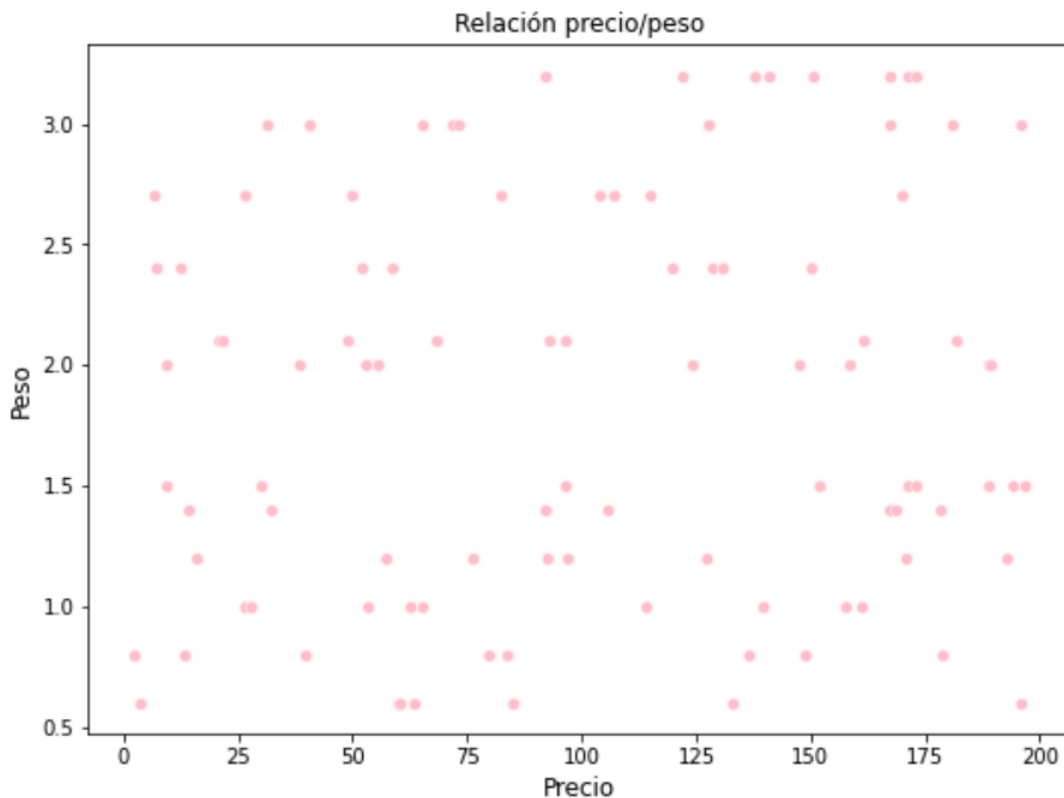
Código Python

```
products['price'] = products['price'].str.replace('$', '')
products['price'] = products['price'].astype('float')
sns.scatterplot(data=products,
                x='price',
                y='weight',
                color="pink",
                sizes=(20, 200))
plt.title("Relación precio/peso");
```

Código para PBI

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
dataset['price'] = dataset['price'].str.replace('$', '')
dataset['price'] = dataset['price'].astype('float')
sns.scatterplot(data=dataset,
```

```
x='price',  
y='weight',  
color="pink",  
sizes=(20, 200))  
plt.xlabel('Precio', fontsize=12)  
plt.ylabel('Peso', fontsize=12)  
plt.title("Relación precio/peso")  
plt.show();
```



- Ejercicio 3

Una variable categórica.

Marcamos en datos "country" y "company_id".

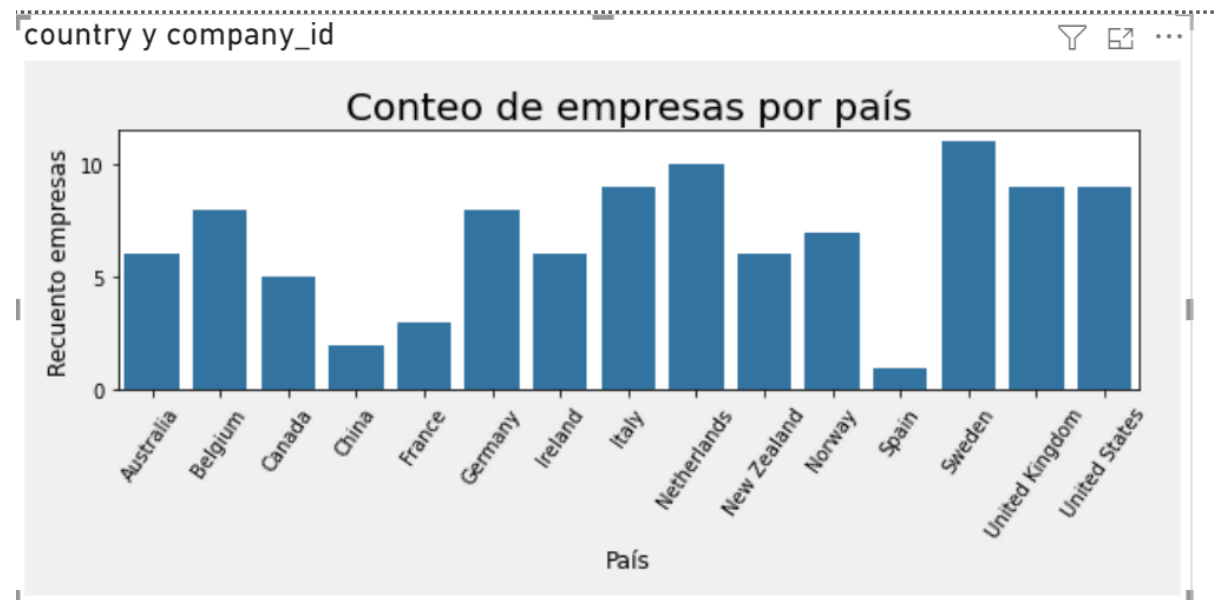
Código Python:

```
sns.countplot(data=companies, x='country')  
plt.style.use('fivethirtyeight')  
plt.title("Conteo de empresas por país")  
plt.xticks(rotation=55) #rotamos las etiquetas del eje X 55 grados para  
#facilitar la lectura porque sino se amontonan.  
plt.tight_layout() #prevenimos que las etiquetas o el título se solapen  
#y se vea ordenado.  
plt.xlabel('País', fontsize=12)  
plt.ylabel('Recuento empresas', fontsize=12)
```

```
plt.show()
```

Código PBI:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(data=dataset, x='country')
plt.style.use('fivethirtyeight')
plt.title("Conteo de empresas por país")
plt.xticks(rotation=55) #rotamos las etiquetas del eje X 55 grados para
#facilitar la lectura porque sino se amontonan.
plt.tight_layout() #prevenimos que las etiquetas o el título se solapen
#y se vea ordenado.
plt.xlabel('País', fontsize=12)
plt.ylabel('Recuento empresas', fontsize=12)
```



- Ejercicio 4

Una variable categórica y una numérica.

Seleccionamos en datos "country" y "amount".

Código en Python:

```
df_mix = pd.merge(companies, transactions, left_on='company_id',
right_on='business_id')
#realizamos un método merge que es como un join en mysql para
relacionar tablas entre sí.
plt.figure(figsize=(12, 6))
sns.boxplot(x='country', y='amount', data=df_mix)
plt.title('Transacciones por país', fontsize=16)
plt.xlabel('País', fontsize=12)
```

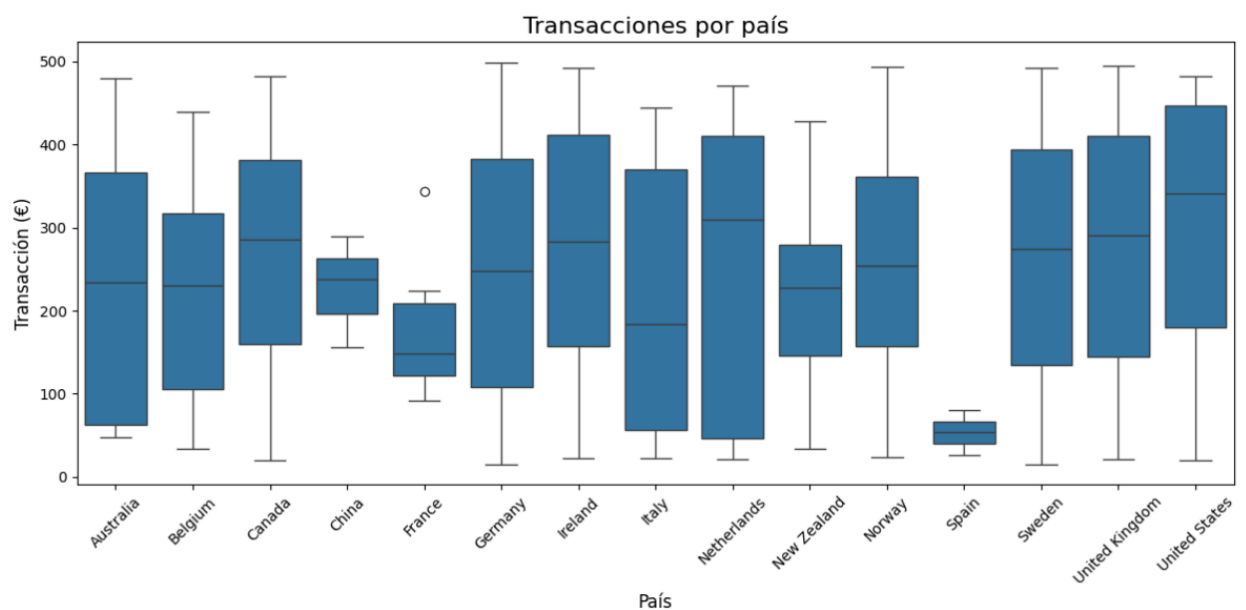


```
plt.ylabel('Transacción (€)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show();
```

Código en PBI:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
plt.figure(figsize=(12, 6))
sns.boxplot(x='country', y='amount', data=dataset )
plt.title('Transacciones por país', fontsize=16)
plt.xlabel('País', fontsize=12)
plt.ylabel('Transacción (€)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show();
```

country y amount



- Ejercicio 5

Dos variables categóricas.

Marcamos en datos “country” de la tabla “companies” y “id” y “declined” de la tabla “transactions”.

Código Python:

```
# Convertir 'declined' en categórica
df_mix['declined'] = df_mix['declined'].astype(int) # Asegúrate de que
sea un entero
df_mix['estado_transaccion'] = df_mix['declined'].map({0: 'Aprobada',
1: 'Rechazada'}).astype('category')
```

```
# Contar transacciones rechazadas por país
transacciones_rechazadas = df_mix[df_mix['declined'] ==
1].groupby('country').size().reset_index(name='count')

# Mostrar el conteo de transacciones rechazadas por país
print(transacciones_rechazadas)

# Gráfico pie chart
plt.style.use('fivethirtyeight')
plt.figure(figsize=(10, 10))
plt.pie(transacciones_rechazadas['count'],
labels=transacciones_rechazadas['country'], autopct='%1.1f%%',
startangle=140)
centre_circle = plt.Circle((0, 0), 0.70, fc='white') # Círculo central
para crear el efecto de dona
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.title('Transacciones Rechazadas por País')
plt.axis('equal')
plt.tight_layout()

plt.show();
```

Código en PBI:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# Convertir 'declined' en categórica
dataset['declined'] = dataset['declined'].astype(int) # Asegúrate de
que sea un entero
dataset['estado_transaccion'] = dataset['declined'].map({0: 'Aprobada',
1: 'Rechazada'}).astype('category')

# Contar transacciones rechazadas por país
transacciones_rechazadas = dataset[dataset['declined'] ==
1].groupby('country').size().reset_index(name='count')

# Mostrar el conteo de transacciones rechazadas por país
print(transacciones_rechazadas)

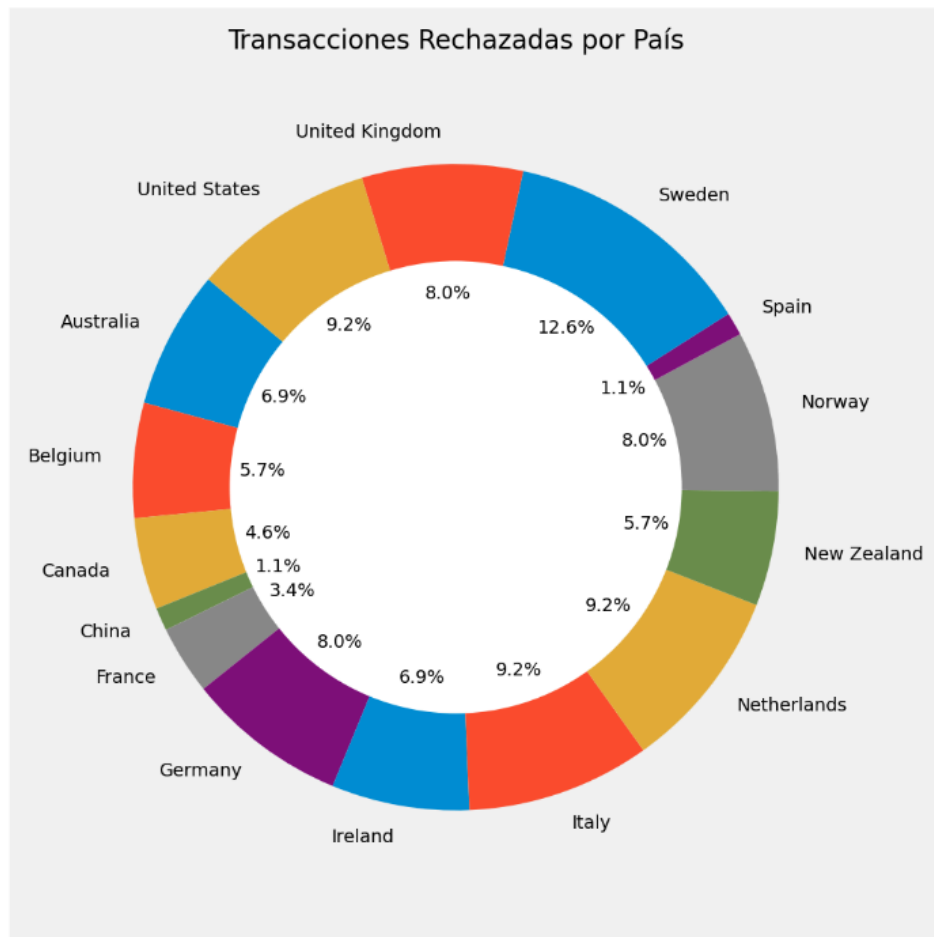
# Gráfico pie chart
plt.style.use('fivethirtyeight')
```

Power BI con Python

Cristina Cumplido Huertas

```
plt.figure(figsize=(10, 10))
plt.pie(transacciones_rechazadas['count'],
labels=transacciones_rechazadas['country'], autopct='%1.1f%%',
startangle=140)
centre_circle = plt.Circle((0, 0), 0.70, fc='white') # Círculo central
para crear el efecto de dona
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.title('Transacciones Rechazadas por País')
plt.axis('equal')
plt.tight_layout()
plt.show();
```

country, declined y id



- Ejercicio 6

Tres variables.

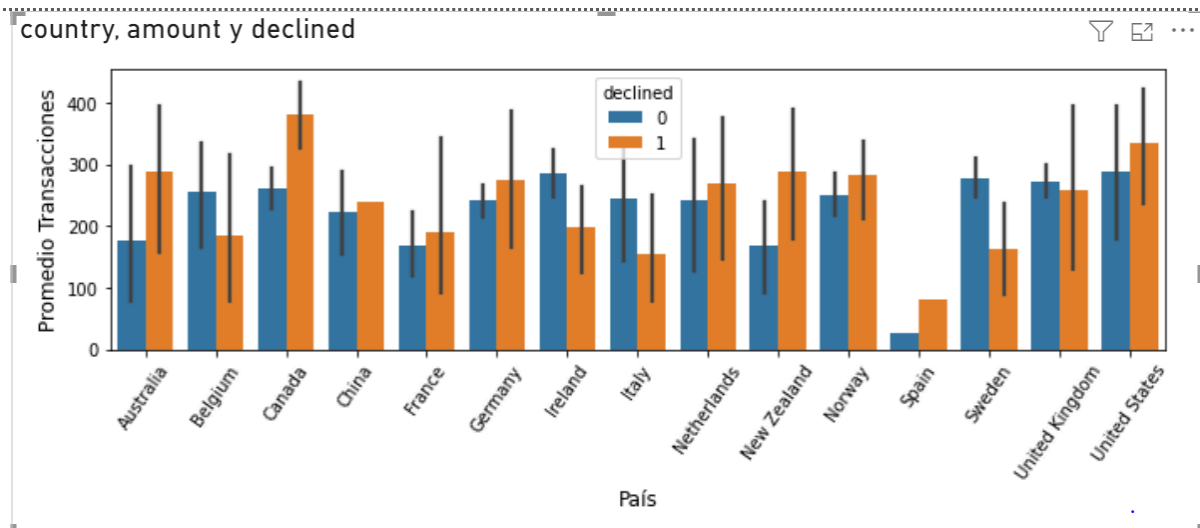
Marcamos en datos “country” de la tabla companies, “amount” y “declined” de la tabla “transactions”.

Código en Python:

```
#barplot te da el amount del país
#vuelvo a poner el merge porque en el ejercicio anterior he modificado
el df_mix
df_1 = pd.merge(companies, transactions, left_on='company_id',
right_on='business_id')
sns.barplot(data=df_1,
            x='country',
            y='amount',
            hue='declined')
plt.xlabel('País', fontsize=12)
plt.ylabel('Promedio Transacciones', fontsize=12)
plt.xticks(rotation=55)
plt.tight_layout();
```

Código en PBI:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
sns.barplot(data=dataset,
            x='country',
            y='amount',
            hue='declined')
plt.xlabel('País', fontsize=12)
plt.ylabel('Promedio Transacciones', fontsize=12)
plt.xticks(rotation=55)
plt.tight_layout()
plt.show();
```



- Ejercicio 8

Graficar un Pairplot.

Marcamos en datos "weight" y "price" de la tabla "products".

Código en Python

```
sns.pairplot(data=products, plot_kws={'color': aqua_palette[4]})  
plt.show();
```

Código en PBI:

```
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
dataset['price'] = dataset['price'].str.replace('$', '')  
dataset['price'] = dataset['price'].astype('float')  
aqua_palette = ["#00FFFF", "#00CED1", "#20B2AA", "#40E0D0", "#48D1CC",  
"#5F9EA0"]  
sns.pairplot(data=dataset, plot_kws={'color': aqua_palette[4]})  
plt.show();
```

price y weight

